Comparative Study and Analysis of Uninformed Search Algorithms

Srajan Jaiswal (Roll No. R134218170)
Amandeep Singh (Roll No. R134218203)
Aman Saraogi (Roll No. R134218204)
Raghvendra Singh (Roll. No R134218128)

Dr. Manoj Kumar
AP - SS
Department of Systemics

ABSTRACT

The search process is a basic task in every computer application wherein the state space search is one of the approaches to handling the job. It intends to find all possible states for a given problem with the desired properties which acts upon the given set of states. Informed search and Uninformed search are the two broad categories of state-space search, this project is based upon the searching techniques placed under the second category of state-space search, The Uninformed search. The two most valuable resources for any optimizing algorithms are time and memory. This project will be implemented taking these two as the base and an optimized algorithm will be crafted.

Keywords

State-space search, Uninformed search, Optimization.

1. INTRODUCTION

Addressing the facet of a problem-solving approach, one is below a specific situation and needs to be within the desired scenario, thus what comes out because the task is to make a series of selections or the series of moves which can remodel the given situation into the required scenario. a hunt may be performed in many various spaces because the given state and therefore the goal state likewise, many alternative states lying in between the 2, the given and therefore the goal state. There are two types of state-space searches, informed search and uninformed search. Area of interest of this project lies around uninformed search under state-space search, which is also known as Blind search as it is unaware of the search space and can only distinguish between a goal state or a nongoal state. Under some circumstances, however, solely a criterion is Goal that is given as a type of the Boolean objective perform. the strategy apart from, that this approach follows, will then not be ready to descend a gradient any longer and degenerate to random walks. Here, the uninformed search is a viable alternative since they do not require or take into consideration any knowledge about the special nature of the problem (apart from the knowledge represented by the expand operation, of course).

Such algorithms are very general and can be applied to a wide variety of problems. Their common drawback is that search areas are usually terribly giant. Their common drawback is that search spaces are often very large. Without the incorporation of information in the form of heuristic functions, for example, the search may take very long and quickly becomes infeasible. Various algorithms come under uninformed searches like Breadth-First Search, Depth First Search, Depth Limited Search, Depth First Iterative Deepening Search, Random Walks, and Bidirectional Search. This Project takes into consideration, four major algorithms that are Breadth-First Search, Depth First Search, Depth-Limited Search, and Depth First search with Iterative Deepening.

2. OBJECTIVES

To perform a comparative study of different performance parameters in Uninformed Search Algorithms listed below to identify and implement the best search approach under different parametric conditions of windows (x32 bit, x64 bit).

1. Breadth-First Search

2. Depth First Search

3. Depth-Limited Search

4. Iterative Deepening Depth First Search

The project aims to overcome the above-stated problems by analysis as well as fair execution. The Iterative Deepening Depth First Search is performed to find the level which then passed to Depth-Limited Search for optimal search.

3. METHODOLOGY

Literature review, detailed study of different search algorithms under Uninformed Search, detailed study of the selected Optimization algorithm, designing process of flow diagram and algorithms, development and Implementation, testing, comparative analysis of algorithms on different environments like x64 bit, report writing.

4. PROBLEM STATEMENT

Searching is carried out to retrieve information stored within the massive data structures, here efficiency becomes a major issue. There exist certain problems which tend to hinder the efficiency of the search algorithms which are taken under consideration in this project.

These Problems are: -

1.The searching of the node in DFS algorithm works well but when the target node lies on the right side of the tree the result will not be optimal.

2.The searching of the node in the BFS algorithm overcomes the problem faced in the DFS algorithm but its memory consumption is much greater than DFS.

3.The searching of a node in DLS algorithm overcomes the problem which is faced in both DFS and BFS algorithm by introducing a limit on the level of search but to find that limit the tree should be solved manually.

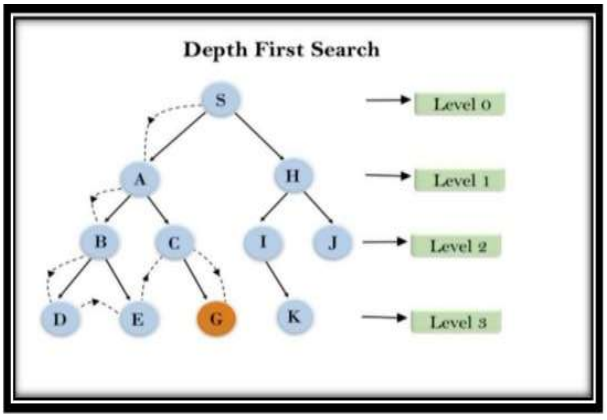
5. RELATED STUDY

Set of all possible states for a given problem is known as State Space of a problem and searching is performed over those state spaces to obtain the desired solution. State-space search is implicit that is the goal nodes is not represented as an explicit object but rather are determined algorithmically from some more concise input.

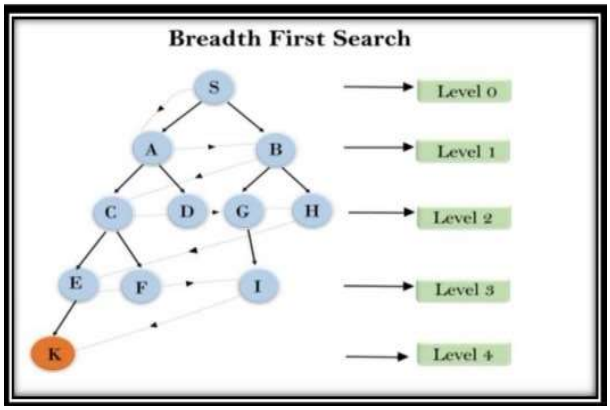
There are two types of state-space searches, informed search and uninformed search. Area of interest of this project lie around uninformed search under state-space search

Uninformed search, which is also known as Blind search as it is unaware of the search space and can only distinguish between a goal state or a non-goal state.

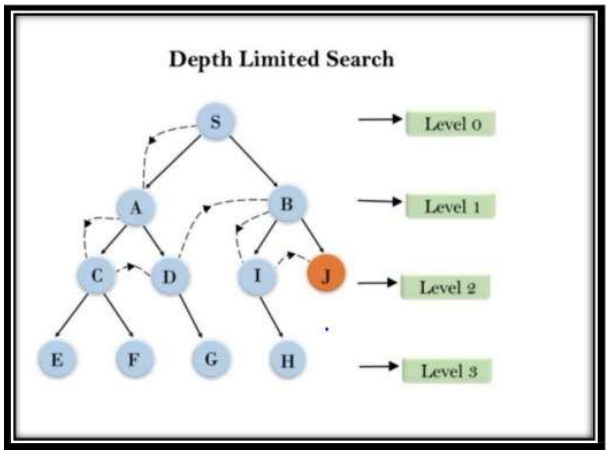
1. Depth First Search: It is a simple search algorithm based on stack implementation. This starts traversing from root explores a path all the way to leaf before backtracking and exploring another path.



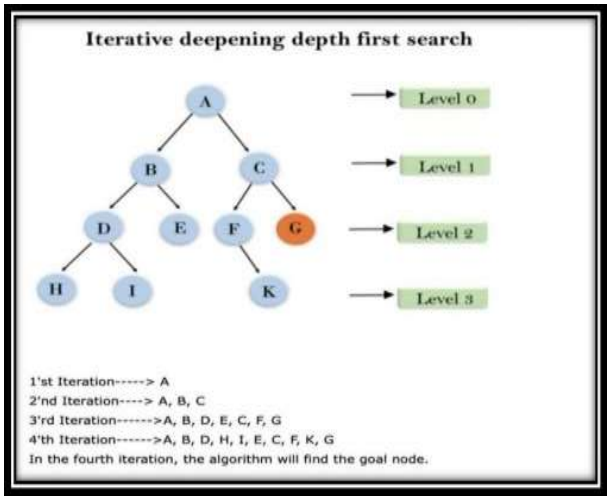
2. Breadth-First Search: It is a simple search algorithm based on queue implementation with a different logic than depth-first search. This traverses the tree level by level and depth by depth. This starts traversing from root explores the neighbour nodes first, before moving to the next node. The number of nodes to be stored in the queue data structure depends on the branching factor of the tree on a particular tree level. It consumes more memory than a depth-first search.



3. Depth Limit Search: This algorithm is an extension of a depth-first search. This algorithm is implemented in a similar way of depth-first search with a slight difference which is, this will be asking the level of the tree along with the destination. With introducing level, it limits the number of comparisons in the tree which is given in limit.

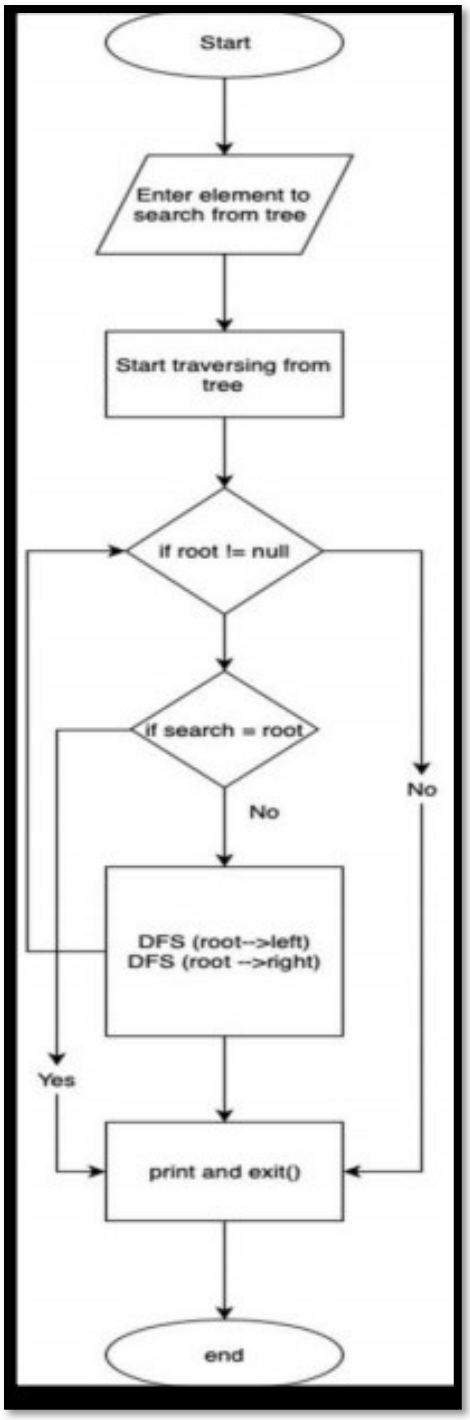


4.Iterative Deepening Depth First Search: It is a combination of depth-first search and breadth-first search. It selects the benefits of both the algorithms and gives out the best solution compared to previous algorithms.

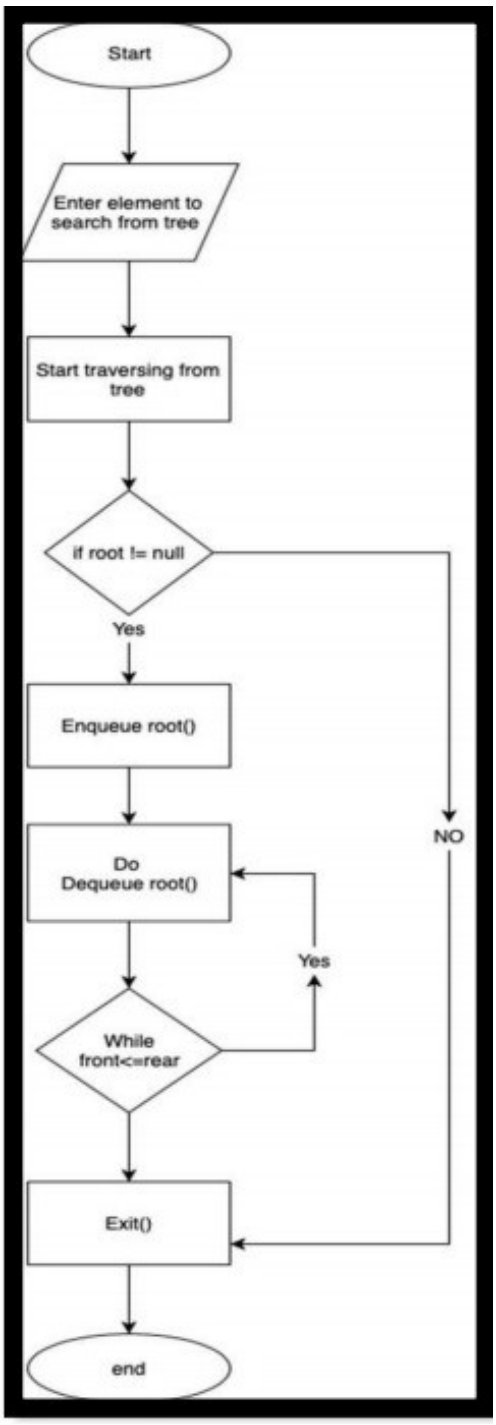


4. ALGORITHM

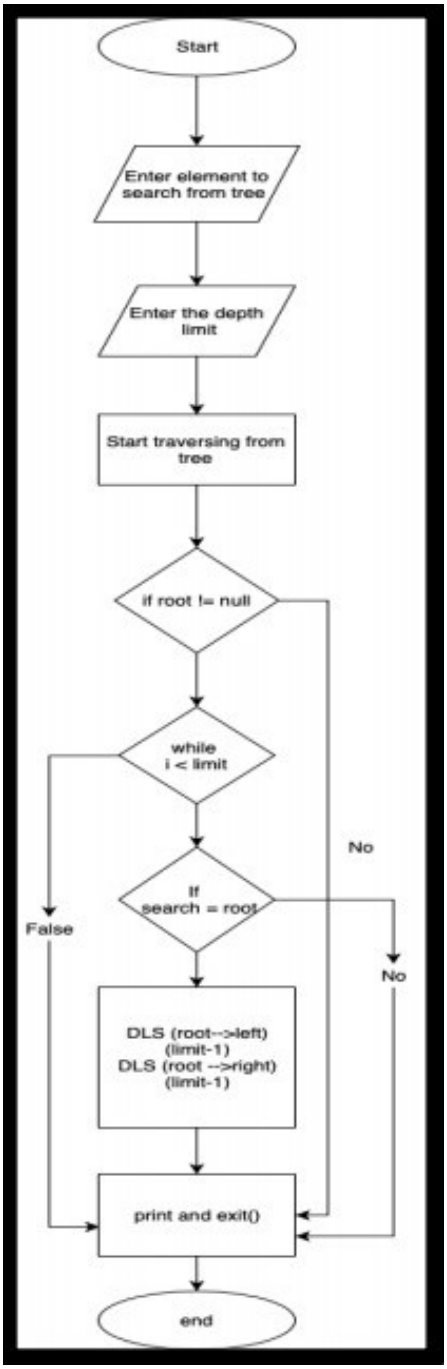
4.1 DFS



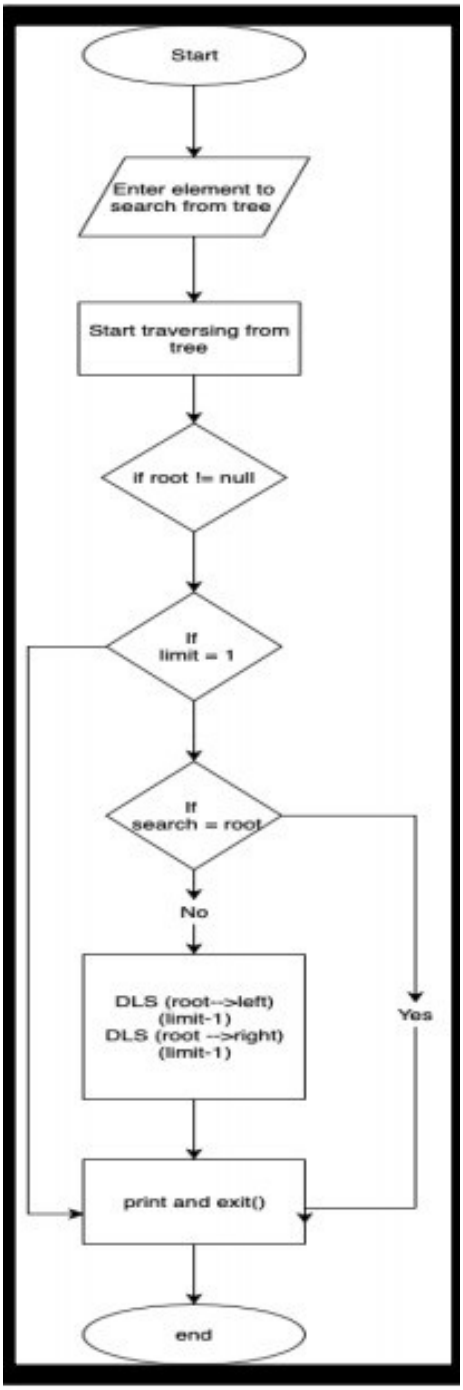
4.2 BFS



4.3 DLS

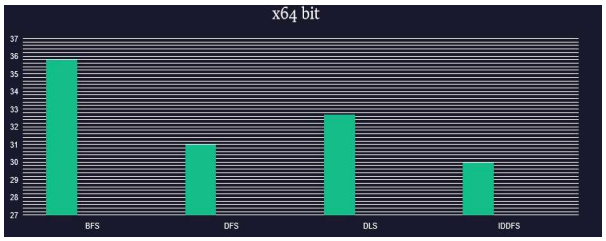


4.4 IDDFS

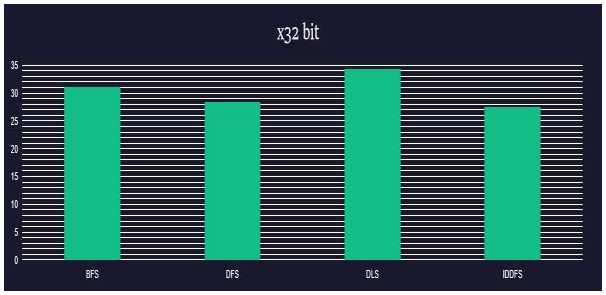


5. RESULT ANALYSIS

Case 1 :-
When the search algorithms are executed in x64 bits of window os, the first set of the result are obtained.
For the total 10 entered values if the value to be searched is ‘9’ the time taken by respective algorithm decreases as we move from BFS to DLS to DFS and finally to IDDFS.



Case 2 :-
When the same search algorithms are executed in x32 bits of windows the second set of the result are obtained.
For the total 10 entered values if the value to be searched is same as case 1 i.e. ‘9’ to provide the same parameter the time taken by respective algorithm again decreases as we move from BFS to DFS.



5. CONCLUSION

We achieved optimization from Depth-First search to Iterative Deepening Depth-First search while taking into consideration each algorithm's time complexities and found the minimum time search algorithm.

This algorithm can be used in many places like: -

1. The algorithm can be used effectively for crawlers in the search engine.

2. The algorithm can also be applied in peer-to-peer networks (Minimum spanning tree).

3. They can be used for Broadcasting in networks.

4. The algorithm can be used for social websites and GPS systems.

6. REFERENCE

[1]https://www.geeksforgeeks.org/graph-and-its-representations/

[2]https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

[3]https://www.geeksforgeeks.org/iterative-deepening-searchids-iterativedeepening-depth-first-searchiddfs/

[4]https://www.geeksforgeeks.org/applications-of-depth-first-search/

[5]https://www.javatpoint.com/ai-uninformed-search-algorithms